

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA

EKONOMICKÁ FAKULTA

KATEDRA SYSTÉMOVÉHO INŽENÝRSTVÍ

Návrh uplatnění servisně orientované architektury v IS podniku

Design of the Service Oriented Architecture Implementation in the Company Information System

Student: Bc. Michal Hanzal

Vedoucí diplomové práce: prof. Ing. Jindřich Kaluža, Csc.

Ostrava 2011

„Místopřísežně prohlašuji, že jsem celou práci včetně všech příloh vypracoval samostatně.“

Podpis:.....

29. dubna 2011

Rád bych touto cestou poděkoval vedoucímu své diplomové práce panu prof. Ing. Jindřichu Kalužovi za velmi cenné připomínky, ochotu a trpělivost po dobu vedení mé práce. Dále bych rád poděkoval panu Ing. Radku Fojtíkovi a panu Ing. Petru Kahánkovi za poskytnutí materiálů, konzultací a cenných rad při realizaci práce.

Obsah

Úvod, cíl práce	- 7 -
Charakteristika společnosti.....	- 8 -
Výroba polovodičů na Moravě.....	- 10 -
Teoretická východiska návrhu servisně orientované architektury	- 12 -
Historie vývoje SOA	- 12 -
Základní vlastnosti současné SOA	- 14 -
Základní komponenty automatizační logiky používané v SOA.....	- 18 -
Základní vlastnosti služeb	- 18 -
Životní cyklus zavádění servisně orientovaných řešení	- 20 -
Servisně orientovaná analýza	- 21 -
Servisně orientovaný návrh	- 21 -
Vývoj služeb	- 21 -
Testování služeb	- 21 -
Zavádění služeb	- 21 -
Správa služeb	- 22 -
Servisně orientovaná analýza	- 22 -
Modelování služeb.....	- 23 -
Servisně orientovaný návrh	- 25 -
Sestavení SOA	- 25 -
Návrh služeb	- 26 -
Shrnutí	- 30 -

Analýza a zhodnocení současného stavu architektury IS podniku	- 32 -
Rozhovor se zaměstnanci	- 32 -
Jednotlivé části řešeného subsystému	- 36 -
Návrh architektury podle servisně orientovaného přístupu.....	- 40 -
Servisně orientovaná analýza	- 40 -
Definice rozsahu analýzy	- 40 -
Identifikace stávajících systémů	- 41 -
Modelování kandidátských služeb.....	- 48 -
Servisně orientovaný návrh	- 52 -
Sestavení SOA	- 52 -
Aplikace standardů	- 52 -
Návrh služby.....	- 52 -
Zhodnocení výsledků zpracovaného řešení.....	- 55 -
Závěr.....	- 56 -
Seznam použité literatury	- 57 -
Seznam zkratk	
Prohlášení o využití výsledků diplomové práce	
Seznam příloh	
Přílohy	

Úvod, cíl práce

Firmy čelící stále se měnícímu obchodnímu prostředí, požadavkům trhů, změnám legislativy a ekonomickým cyklům vyžadují taková podniková řešení, která jim jsou schopny poskytnout co nejefektivnější dosažení podnikových cílů. Pro úspěch firem v dnešním bouřlivě se měnícím ekonomickém prostředí už nestačí jen samotné řízení jejich výkonnosti, klíčovým hráčem na tomto poli se stává schopnost přizpůsobení se. Hledání řešení, která budou schopna podniku přinést jak optimalizaci nákladů, tak schopnosti přizpůsobení se, bude kritické pro firmy, které se budou muset adaptovat na měnící se požadavky svých zákazníků, které budou muset rychle reagovat na nepředvídané události a které si chtějí udržet funkční růstové strategie i v obdobích ekonomických recesí. Je jasné, že se nedá do nekonečna spoléhat na přísun stále většího množství zdrojů, je třeba začít pracovat „chytřeji“, lépe využívat to, co už máme.

Všechny výše zmíněné požadavky a mnohé další je schopen v podobě integrace v informačním systému podniku poskytnout servisně orientovaný přístup k budování architektury informačního systému.

Cíle této diplomové práce jsou dva: prvním je jasné a přesné vymezení a objasnění pojmu servisně orientovaná architektura, což bude provedeno v kapitole Teoretická východiska návrhu servisně orientované architektury. Druhým cílem je praktické využití teoreticky nastíněného návrhu servisně orientovaných architektur v podobě návrhu implementace SOA v informačním systému společnosti ON Semiconductor.

Charakteristika společnosti

Jak už bylo uvedeno v předchozí části této práce, budou principy návrhu architektury informačních systémů pomocí servisně orientovaného přístupu implementovány v nadnárodní společnosti ON Semiconductor.

Vznik společnosti ON Semiconductor se datuje k 1. srpnu roku 1999, kdy se od korporace Motorola, Inc. odštěpila Semiconductor Components Group. Hlavní ústředí firmy se nachází ve městě Phoenix ve státě Arizona v USA a po celém světě je rozmístěno na 45 poboček (v České republice jsou to centra v Brně a Rožnově pod Radhoštěm), které jsou rozděleny na návrhová a výrobní střediska. Celkový počet zaměstnanců korporace se pohybuje okolo 13.000¹.

Zaměření společnosti ON Semiconductor je návrh a výroba čipů a polovodičových součástek. Hlavními oblastmi průmyslu, kde je možné součástky ON Semiconductor, zobrazuje následující tabulka.

Tabulka 1 - Uplatnění součástek ON Semiconductor, Zdroj: firemní materiály

Odvětví	Výrobky	Partnerské firmy
Automobilový průmysl	<ul style="list-style-type: none">• osvětlení• zabezpečovací mechanismy• audio• řídicí jednotky	Siemens, Bosch, Continental
Výpočetní technika	<ul style="list-style-type: none">• notebooky• PC pracovní stanice• servery• herní zařízení, grafické adaptéry	Lenovo, HP, NVidia, Intel, ASUS, Dell

¹ Zdroj: www.onsemi.com

	<ul style="list-style-type: none"> • komunikační zařízení 	
Spotřební elektronika	<ul style="list-style-type: none"> • LCD displeje • DVD přehrávače • Set-Top boxy 	Sony, LG, Microsoft, Motorola, Samsung
Přenosná zařízení	<ul style="list-style-type: none"> • mobilní telefony • PDA • digitální kamery • GPS navigační systémy • MP3 přehrávače 	Sony, Motorola, Apple, Garmin, Kodak, Sony Ericsson
Medicína	<ul style="list-style-type: none"> • zobrazovací zařízení • implantační zařízení • diagnostika 	Medtronic, GE Medical, Boston Scientific
Vojenství, letectví, kosmonautika	<ul style="list-style-type: none"> • kokpitové displeje • navigační systémy • infračervené zobrazování • vojenské komunikační systémy 	Honeywell, Flir Systems, Raytheon, Rockwell Collins
Sítě	<ul style="list-style-type: none"> • optické sítě • routery a switche • bezdrátové stanice 	Cisco, Nokia-Siemens Networks, Ericsson, Huawei, Broadcom, Alcatel-Lucent

Výroba polovodičů na Moravě²

Tradice výroby polovodičů a polovodičových součástek v Rožnově pod Radhoštěm je spjata s podnikem Tesla, který zde již ve 40. letech minulého století začal s výrobou elektronek a elektronkových obrazovek a později v letech 50. pokračoval návrhem a výrobou prvních polovodičových součástek (i přes zjevný odpor tehdejší vlády). Postupně pak tehdejší Tesla Rožnov přešla k výrobě logických obvodů, v čemž pokračovala až do 80. let minulého století, výsledkem práce mnoha inženýrů jsou pak některé přelomové návrhové vzory integrovaných obvodů, jejichž variace jsou v hojné míře používány různými podniky dodnes.

Tesla Rožnov byla považována za jeden z mála Československých podniků, který byl za komunistické éry schopen do jisté míry konkurovat západoevropským podnikům a prakticky byla tahounem střední a východní Evropy v návrhu a výrobě polovodičových součástek a integrovaných obvodů. Toto ovšem nezabránilo událostem na přelomu 90. let, aby kolosem, jakým tehdy Tesla Rožnov byla, těžce otřásly. Zhroucení východního trhu a dovoz levných součástek vedly k tomu, že Tesla Rožnov nebyla ve své tehdejší struktuře a se svým sortimentem schopna uplatnit se na světovém trhu. Až do roku 1994 o přežití bojovaly firmy Tesla SEZAM a TEROSIL jako nástupkyně rozpadnuvší se Tesly Rožnov; v tomto období začala postupná spolupráce se společností Motorola, Inc. Ve stejném roce je rovněž založeno středisko Czech Design Center (CDC), které plnilo a dodnes plní funkci předního návrhového střediska v rámci společnosti ON Semiconductor, jak v oblasti návrhu polovodičových součástek, tak návrhu a vývoje softwarových aplikací. V roce 1999 došlo ke vzniku společnosti ON Semiconductor oddělením od společnosti Motorola, Inc.

V rámci CDC v Rožnově pracují tři hlavní výzkumná a vývojová střediska³:

- New Product Development (NPD) – vývoj a uvedení do výroby nových integrovaných obvodů

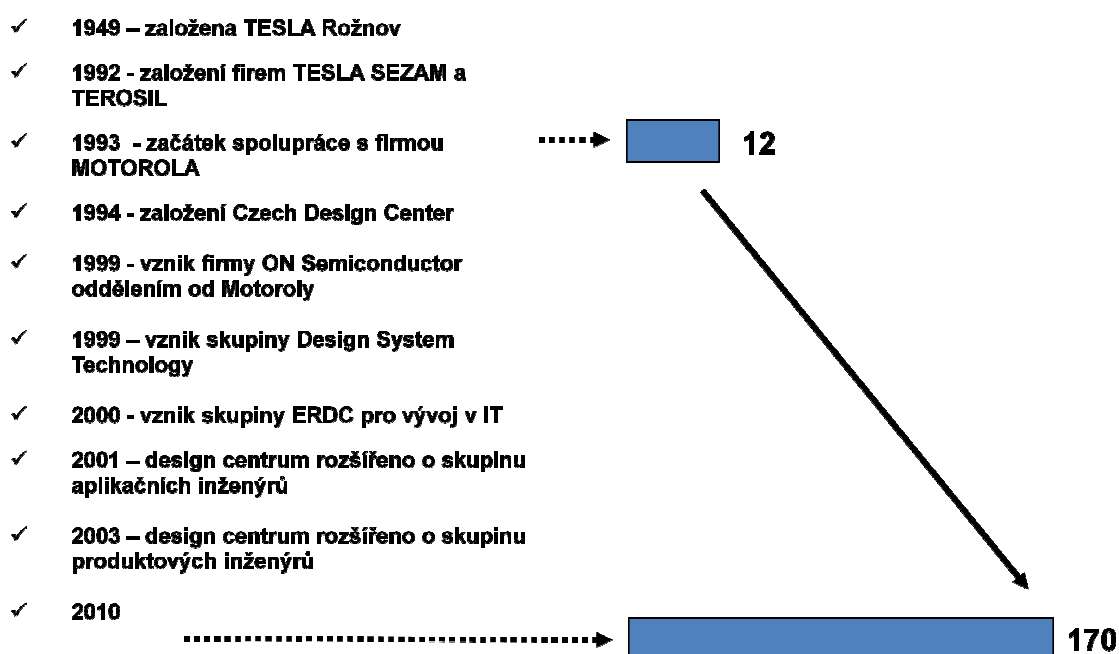
² Zdroj: Baltus, 2010a a Baltus, 2010b

³ Zdroj: <http://www.onsemi.com/PowerSolutions/content.do?id=15001>

- Design System Technologies (DST) – vývoj návrhových knihoven a nástrojů pro charakterizaci procesů a tvorbu modelů součástek
- European Region Development Center (ERDC) – vývoj globálních firemních softwarových produktů a podpora projektů v této oblasti

Obrázek 1 - Hlavní milníky historie výroby polovodičových součástek v Rožnově ilustruje hlavní milníky historie výroby polovodičových součástek v Rožnově pod Radhoštěm⁴.

Historie



Obrázek 1 - Hlavní milníky historie výroby polovodičových součástek v Rožnově, Zdroj: firemní materiály

Tato diplomová práce vznikla za podpory střediska analýzy a vývoje ERDC v Rožnově pod Radhoštěm.

⁴ modré obdélníky a čísla znázorňují vývoj počtu zaměstnanců

Teoretická východiska návrhu servisně orientované architektury

Následující část se bude podrobně zabývat pojmem servisně orientovaná architektura (SOA). Tento pojem bude vysvětlen, bude zde popsána stručná historie vzniku SOA a jejího postupně pokračujícího vývoje, budou stanoveny základní vlastnosti SOA a základní vlastnosti webových služeb. Dále se budeme zabývat životním cyklem zavádění SOA v podniku. A rovněž bude rozebrána problematika servisně orientované analýzy a návrhu a představeny dvě klíčové technologie spojené se servisně orientovanou architekturou, jazyk WSDL a komunikační protokol SOAP. Jelikož je SOA poměrně novým přístupem k budování informačních systémů, není v současné době ještě dostupný dostatek odborné literatury, která by tuto problematiku zevrubně rozebírala. Proto byla jako hlavní zdroj teoretických informací použita publikace Thomase Erla (Erl, 2009), kterou jsem se snažil podle možností doplnit zejména internetovými články.

Historie vývoje SOA

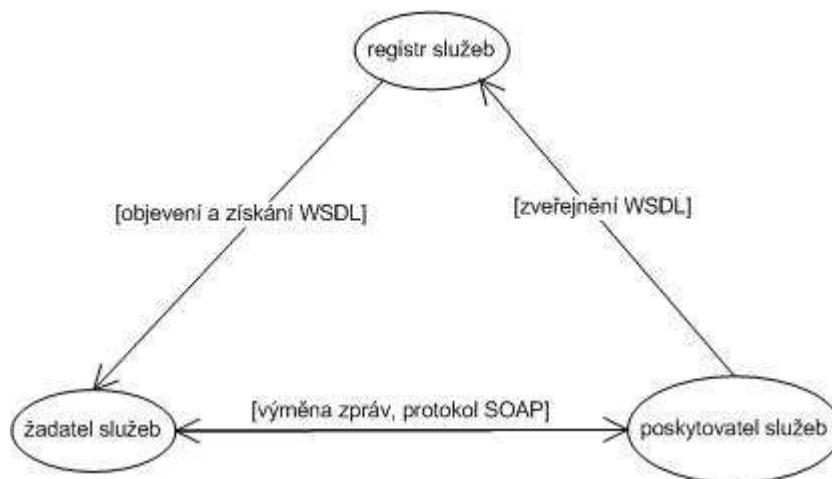
Za vznikem servisně orientovaných architektur stojí hned několik technologií, z nichž nejpodstatnější jsou jazyk *XML* (Extensible Markup Language) a *webové služby*.

XML se pod záštitou W3C⁵, podobně jako některé další jazyky, vyvinul z jazyka SGML (Standard Generalized Markup Language), který byl vytvořen v 60. letech minulého století. Jazyk XML se začal výrazně rozšiřovat v průběhu 90. společně s postupným přesouváním obchodu na internet; jeho hlavní předností je to, že přiřazuje strukturu a význam elektronickým dokumentům, potažmo informacím přenášeným internetovými protokoly. Ve vztahu k servisně orientované architektuře, představuje XML její elementární vrstvu, neboť definuje formát, strukturu a význam zpráv přenášených mezi jednotlivými službami. Jak uvádí Erl (2009, str. 70): „V SOA doslova neuděláte krok bez toho, aniž byste použili XML.“

⁵ World Wide Web Consortium

Původcem vzniku technologie webových služeb je specifikace protokolu SOAP (Simple Object Access Protocol), která byla přijata W3C v roce 2000. Princip protokolu je takový, že informace přenášené mezi dvěma prvky se převedou do XML, přenesou a následně zpětně převedou do původního formátu. Samotná technologie webových služeb vznikala na základě vývoje na poli internetového obchodu. Jednotlivé firmy se tak dlouho snažily odstranit nejednotnosti, které mezi nimi existovaly, až se zrodila myšlenka na vytvoření čistě webové distribuované technologie; od tohoto okamžiku už byl jen krůček k první specifikaci webových služeb. Protokol SOAP byl pak vybrán jako hlavní protokol pro použití s webovými službami. Důležitou součástí technologie webových služeb je jazyk WSDL (Web Service Definition Language), který poskytuje model pro popis jednotlivých služeb ve formátu XML.

Postupem času společnosti, které implementovaly webové služby do svých podnikových řešení, zjišťovaly, že by mohly být využity jako fundament samostatných architektonických platforem. Tímto okamžikem se servisně orientovaná architektura dostala do popředí zájmů odvětví informačních technologií a začíná její rapidní rozvoj. Sloučení technologií XML a webových služeb s podporou registru služeb (tak jak je znázorněno na Obrázek 2 - Prvotní SOA, Zdroj: Erl, 2009) do základu samostatné architektonické platformy se dnes nazývá *prvotní SOA*. Servisně orientovaná architektura se stále vyvíjí, a proto se může stát, že cokoli je dnes považováno za čisté SOA, může být v příštích letech zavrhnuto. Není proto ani jednoduché definovat základní principy současné SOA, Thomas Erl se ovšem touto problematikou ve své knize podrobně zabývá a tak i v následující části druhé kapitoly se pokusím nastínit v dnešní době globálně přijímané principy servisně orientované architektury.



Obrázek 2 - Prvotní SOA, Zdroj: Erl, 2009 a podobně Barry, 2011 nebo Weiss a Rychlý, 2007

Základní vlastnosti současné SOA

V první řadě je nutné vysvětlit pojem *současná SOA*. Je to takový model SOA, který je postaven na principech prvotní SOA a do současné podoby byl doveden vývojem. Hlavní vlastnosti současné SOA, jak uvádí Erl (2009) a doplňuje The Open Group (2010), jsou následující:

- Je součástí jádra servisně orientované výpočetní platformy – tento predikát zakládá na tom, jak vlastně vznikl název servisně orientovaná architektura a poťazmo jeho zkratka SOA v komunitě zabývající se informačními technologiemi. V minulosti i současnosti se setkáváme se spoustou termínů označujících aplikační výpočetní báze, které obsahují slovo architektura (např. architektura klient-server). Tyto infrastruktury mohou být založeny na nepřehledném množství technologií, naopak cokoli, co je označeno jako SOA, je vytvořeno podle servisně orientovaných principů.
- Zlepšuje kvalitu služby – tato vlastnost obsahuje tradiční architektonické požadavky, jako jsou provádění úloh bezpečným způsobem, ochrana obsahu přenášených zpráv, ošetření výjimečných stavů, požadavky na výkon, transakční schopnosti spojené s ochranou integrity řídicích úloh, jejichž selhání spouští výjimečnou logiku apod.,

- Je zásadně autonomní – principy servisní orientace vyžadují, aby jednotlivé služby byly tak nezávislé a autonomní, jak to jen bude možné. Toho je dosaženo tak, že zprávy, které si vyměňují jednotlivé služby, jsou dostatečně inteligentní na to, aby kontrolovaly způsob, jakým mají být zpracovány přijímajícími službami.
- Je založena na otevřených standardech – veškerá výměna dat a komunikace v rámci současné SOA je řízena výhradně otevřenými standardy; tzn., že každá zpráva, kterou si služby vyměňují je přenášena sadou otevřených, obecně přijímaných standardů a navíc i samotné zprávy jsou standardizovány jak ve svém formátu, tak ve způsobu, jakým charakterizují svůj obsah. Oněmi používanými otevřenými standardy v rámci SOA jsou SOAP, WSDL a XML.
- Podporuje rozmanitost prodejců – v rámci současné SOA nezáleží na tom, kdo je výrobcem vývojové platformy, nebo jakou technologii tato platforma využívá. Je potom možné, aby spolupracovaly navzájem neslučitelná firemní řešení, jako například .NET a J2EE.
- Stará se o vnitřní spolupráci – bez ohledu na to zda navrhovaná aplikace nebo podnikové řešení vyžaduje integraci, je možné služby vybavit vlastnostmi, které přirozeně podporují spolupráci. Aplikace složené z takto vybavených samostatných služeb pak samy sebe vnímají jako složené autonomní služby.
- Podporuje zjistitelnost – i když v definici rané SOA je uveden registr služeb jako nedílná součást celého konceptu, nebyly v minulosti registry služeb použity jako součásti podnikových řešení. To mohlo být dáno faktem, že fungovalo relativně malé množství služeb v tradičních architektonických řešeních, jejichž primárním úkolem bylo je usnadňovat komunikaci. Současná SOA silně doporučuje použití zavedení určité formy řízení registrace služeb.
- Podporuje federace – implementace SOA řešení do informačních systémů podniků nemusí rovnou znamenat nahrazení současných, často i mnoho let spolehlivě fungujících řešení. Právě naopak je SOA schopna přinést jednotnost do dříve neslučitelných aplikací, tím, že je možné zapouzdřit zastaralou aplikační logiku a použít ji pomocí dříve definovaného, otevřeného systému komunikace.
- Podporuje princip architektonické kompozice – kompozice v rámci SOA je jednou z klíčových vlastností a lze ji provádět na různých úrovních. Jak už bylo ře-

čeno dříve, jsou služby navrhovány jako autonomní a je brán zřetel na jejich znovupoužitelnost; služby tedy lze komponovat do složitých struktur.

- Stará se o základní znovupoužitelnost – služby navrhované podle principů servisně orientované architektury by měly být automaticky navrhovány tak, aby byla možnost jejich znovupoužití, i když neexistuje přímá potřeba jejich opětovného použití. Pokud je toto pravidlo dodrženo, vzniká systém, v němž je znovupoužití realizováno jako přirozená vedlejší funkce.
- Zdůrazňuje rozšiřitelnost – služby lze rozšiřovat a přidávat jim tak další požadovanou funkcionalitu tak, aniž by bylo měněno současné komplexní řešení. Rozšíření se realizuje přidáváním dalších služeb nebo sloučením několika servisně orientovaných aplikací.
- Implementuje vrstvy abstrakce – vrstvy abstrakce mohou být v servisně orientovaných řešeních zavedeny vytvořením služeb jako výlučných přístupových bodů k různým aplikacím nebo subsystémům.
- Podporuje volnou vazbu v celém podniku – princip volné vazby spočívá v nezávislosti služeb; službám stačí, aby se navzájem uvědomovaly, což následně umožňuje jejich další nezávislý vývoj. Jednotlivým aplikacím a subsystémům pak stačí, aby znaly své „koncové“ body a „koncové“ body aplikací a subsystémů, se kterými komunikují, není nutné, aby navzájem znaly své vnitřní struktury.
- Podporuje organizační flexibilitu – dříve uvedené principy znovupoužitelnosti, rozšiřitelnosti, abstrakce a volných vazeb podporují v podniku schopnost reagovat na vyvstanuvší požadavky na změny, které mohou vyvstat například při reorganizaci, sloučení společností, nebo při změně oboru podnikání a podobně.
- Je stavební blok – zavádění SOA v podniku, by mělo vést k ideální situaci, která je označována jako Service Oriented Enterprise (SOE), neboli servisně orientovaný podnik. V takovéto organizaci potom servisně orientovaná architektura plní funkci základní stavební architektury a vytvořené služby, aplikace apod. jsou základními stavebními bloky.
- Je vývoj – servisně orientovaná architektura přejala od tradičních distribuovaných prostředí a prostředí klient-server jejich úspěšné vlastnosti (jako jsou znovupoužitelnost, zapouzdření, komponovatelnost a další), ale používá odlišné pro-

středky analýzy a návrhu, které jsou ovlivněny servisní orientací a webovými službami.

- Je dosažitelný ideál – přijetí SOA jako standardu celé organizace znamená proces přeměny podniku do stavu, který už byl zmíněn výše, a to do stavu SOE. Takovýto přechod znamená vynaložení obrovského úsilí a v závislosti na velikosti podniku také financí a času. Každé technické řešení časem podstoupí určité změny v kontextu servisní orientace, což jistě povede ke stavu, kdy se v podniku bude vyskytovat množství hybridních řešení, tzn. takových, kde již budou zavedeny prvky servisní orientace, ale stále budou v jádru postavena na starých distribuovaných řešeních.

Nyní je možné z výše popsaných základních vlastností současné SOA sestavit její definici, tak jak ji uvádí Erl (2009, str. 55):

„Současná SOA představuje otevřenou, rozšiřitelnou, federační a komponovatelnou architekturu, která podporuje servisní orientaci a skládá se ze služeb, které jsou autonomní, schopné QoS, podporují rozmanitost prodejců, spolupráci, zjistitelnost, a jsou implementované jako webové služby. SOA může zavést abstrakci řídicí logiky a technologie, což vede k volné vazbě mezi těmito doménami. SOA se vyvinula ze starších platform, zachovává užitečné vlastnosti tradičních architektur a přichází s různými principy, které se starají o servisní orientaci na podporu servisně orientovaného podniku. SOA je ideálně univerzálním standardem v rámci celého podniku, ale dosažení tohoto stavu vyžaduje plánovaný přechod a podporu stále se vyvíjející sady technologií.“

Tato definice je příliš dlouhá, těžkopádná a všeobsahující, proto ji Erl zkrátil do této podoby (Erl, 2009, str. 56; podobně definuje i Barry, 2011):

„SOA je forma technologické architektury, která lpí na principech servisní orientace. Pokud je realizována prostřednictvím technologické platformy webových služeb, SOA vytváří potenciál pro podporu a vylepšení těchto principů pomocí řídicího procesu a automatizace domén podniku.“

Základní komponenty automatizační logiky používané v SOA

SOA a respektive systém služeb neposkytuje jen technologickou bázi procesu tvorby informačních systémů, ale také pohled na modulární složení logiky automatizace z autonomních jednotek.

Za základní komponenty automatizační logiky Erl (2009) označuje:

- zprávy – představují základní jednotky komunikace,
- operace – představují základní jednotky práce,
- služby – představují základní jednotky logiky zpracování (nebo také kolekce jednotek práce),
- procesy – představují jednotky logiky automatizace.

Základní vlastnosti služeb

Obecně neexistuje žádná oficiální sada principů servisní orientace. To, co se za obecně přijímanou sadu principů nejvíce spojených se servisní orientací považuje, jsou následující vlastnosti služeb.

Znovupoužitelnost

Znovupoužitelnost je automaticky podporována, přičemž nezáleží na tom, zda existují přímé požadavky na opětovné použití jednotlivých služeb. Vytvářením služeb, které automaticky splňují tuto vlastnost, se návrhář v budoucnu vyvaruje potíží, jež by mohly nastat při úpravách požadavků na operace služeb.

Sdílení formální dohody

Dohoda služeb definuje jejich koncové body, operace, typy podporovaných vstupních a výstupních zpráv a pravidla a vlastnosti. Formální dohoda tedy poskytuje informaci o sémantice dané služby, neboli jak daná služba vykonává příslušné úlohy. Dohoda služeb je sdílená, a proto je její bezchybný návrh velice důležitý, neboť žadatelé služeb se na ni po přijetí stávají závislími, proto nesmí být dohody po vydání měněny.

Volná vazba

Jako volná vazba mezi službami je označován stav, kdy se služby mohou mezi sebou dozvědět o své existenci, a přesto si zachovat nezávislost. Ke stavu volné vazby se dospěje použitím formálních dohod popsanych v předchozím odstavci.

Abstrakce logiky v pozadí

Tato vlastnost umožňuje službám vůči okolí jako černé skřínky, jejichž detailní vnitřní složení ani logika, kterou zapouzdřují, není známa. Nejsou stanovena omezení pro to, jak velkou část logiky informačního systému může služba zabezpečovat – může být jak branou k celému informačnímu systému, tak také službou, která vykonává jednu jednoduchou úlohu. Vlastnost abstrakce logiky je vlastně důsledkem předchozích vlastností, protože aby bylo dosaženo volné vazby, jediné informace, které o sobě služby navzájem potřebují vědět, jsou dány formálními dohodami.

Komponovatelnost

Jak bude popsáno dále, služby vznikají rozložením firemních procesů, z čehož vyplývá, že proces⁶ je kompozicí jednotlivých služeb, které vykonávají určité části logiky daného procesu. Z jiného pohledu je komponovatelnost jiným projevem vlastnosti znovupoužitelnosti, protože služba, která zapouzdřuje logiku v jednom procesu, může být znovuvyužita v jiném procesu.

Autonomnost

Požadavek autonomnosti zajišťuje, aby služba byla schopna provádět samostatně veškeré své operace, čímž se služba rovněž zbavuje závislosti na ostatních službách. Rozlišují se dva typy autonomie:

- autonomie na úrovni služeb – situace služby, která zapouzdřuje zastaralou aplikaci, která je využívána i mimo zapouzdřující službu,

⁶ Zde se mluví o procesech, které mohou být automatizovány pomocí aplikace informačních a komunikačních technologií.

- plná autonomie – logika, kterou služba zapouzdřuje, je pod výhradní kontrolu zapouzdřující služby (tato situace nastává, když je logika v pozadí služby od začátku tvořena pro danou službu).

Bezstavovost

Je vhodné, aby služby držely co nejméně informací o svých aktivitách, to znamená, aby byly bezstavové. Vysokého stupně bezstavovosti se dosahuje přesunutím „intelligence“ do zprávy, které jsou mezi službami posílány.

Zjistitelnost

Tato vlastnost se snaží předciť tomu, aby byly tvořeny redundantní služby nebo logika. Každá služba obsahuje potenciálně znovupoužitelný kus logiky, musí popisy služeb obsahovat dostatečné množství informací, aby popisovaly i operace služeb.

Životní cyklus zavádění servisně orientovaných řešení

Proces zavádění servisně orientovaných řešení, podobně jako jiné podobné projekty, zahrnuje určitou sadu kroků, které musí být vykonány, aby byl tento proces doveden do úspěšného konce. Kroky životního cyklu zavádění SOA jsou následující⁷:

- servisně orientovaná analýza,
- servisně orientovaný návrh,
- vývoj služeb,
- testování služeb,
- zavádění služeb,
- správa služeb.

⁷ Zdroj: Erl, 2009 a podobně *Microsoft BizTalk Server*, 2006

Servisně orientovaná analýza

Úvodní fáze životního cyklu, ve které se určuje rozsah aplikace servisně orientované architektury. Servisně orientovaná analýza bude podrobněji popsána v další části této kapitoly.

Servisně orientovaný návrh

Servisně orientovaná analýza odpovídá na otázku „Co se bude budovat?“. Jakmile je tedy jasné, co bude vytvořeno, je nutno si vyjasnit, jak to bude provedeno. Tímto se zabývá servisně orientovaný návrh, který bude podrobněji popsán v části Servisně orientovaný návrh.

Vývoj služeb

Jedná se o fázi samotné konstrukce navrhovaných řešení. Je zde řešen problém výběru vývojové platformy, zejména pak výběr programovacích jazyků, v nichž budou jednotlivé služby naprogramovány, čímž je rozhodnuto o fyzické podobě služeb, což vyplývá z návrhu.

Testování služeb

Tato fáze ověřuje, zda služby splňují základní vlastnosti servisně orientovaných řešení (jako například znovupoužitelnost), ale také jejich faktickou funkčnost a správnost.

Zavádění služeb

V této fázi jsou služby nasazovány do „ostrého“ provozu. Jsou řešeny problémy týkající se spolupráce s již nasazenými aplikacemi, úpravy bezpečnostní politiky systému, ideální rozmístění služeb používaných více aplikacemi systému apod.

Správa služeb

Po zavedení nového řešení nastává doba řešení klasických problémů spojených s údržbou podnikových řešení. Je třeba sledovat, jak jsou služby využívány a případně detekovat slabá místa snižující výkon aplikace, nebo stanovit systém verzování atd.

Je nutno připomenout, že v celém životním cyklu SOA dochází k neustálým iteracím, tzn., že jednotlivé fáze se stále opakují. Pokud je tedy například ve fázi testování služeb odhalena kritická chyba, která by v budoucnu mohla způsobit řadu těžkostí, je nutno se vrátit například až k fázi návrhu a znova celé původní řešení přestavět.

Tato diplomová práce se zabývá zejména prvními dvěma fázemi životního cyklu, tedy servisně orientovanou analýzou a návrhem. Fáze ostatní stojí mimo záběr této práce, a proto již v dalším textu nebudou podrobněji rozpracovány.

Servisně orientovaná analýza

Erl (2009) definuje servisně orientovanou analýzu jako „postup, který rozhoduje o tom, jak lze reprezentovat požadavky na automatizaci řízení prostřednictvím servisní orientace.“ Servisně orientovaná analýza tak odpovídá na dvě základní otázky⁸:

- Jaké služby mají být vytvořeny?
- Jakou logiku má každá služba zapouzdřovat?

Stejně jako životní cyklus, popsáný v minulé části této kapitoly, tak i jeho část servisně orientovaná analýza je vnímána jako proces, který je definován třemi kroky:

- 1) Definice rozsahu analýzy – definování rozsahu projektu je jednou za základních činností. U SOA se obvykle vychází z řídicích podnikových procesů, které musí být jasně formulovány a dopodrobna zdokumentovány. I když je tvořena jen jedna služba, může být rozsah celého projektu velký, neboť zcela závisí na procesu, který bude daná služba zapouzdřovat. Tento proces může být natolik rozsáhlý, že bude vyžadovat vytvoření několika podpůrných služeb a samotná služba, která

⁸ Podobně popisuje postup servisně orientované analýzy také Arsanjani, 2004

proces zapouzdřuje, se může z několika dalších služeb sestávat (tato vlastnost služby se nazývá granularita služby).

- 2) Identifikace stávajících systémů automatizace – v tomto kroku je třeba zjistit, jaké aplikace v informačním systému jsou již zavedeny a jak je zavádění nových servisně orientovaných řešení bude ovlivňovat. Detaily spolupráce starých a nových aplikací jsou následně doladěny ve fázi servisně orientovaného návrhu.
- 3) Modelování kandidátských služeb – tato část procesu servisně orientované analýzy je klíčová, podrobněji jí budeme zabývat v další části této kapitoly.

Modelování služeb

Hned na začátku je nutné si vyjasnit rozdíl mezi službou a kandidátem služby. Ve fázi analýzy se nevytváří návrh informačního systému, ale jsou identifikovány potenciální procesy, ze kterých by mohly být služby ve fázi servisně orientovaného návrhu vytvořeny. Analytická fáze tedy nevytváří služby jako takové, ale pouze jejich kandidáty. Výsledkem procesu modelování služeb jsou potom kandidáti služeb a kandidáti operací služeb.

Modelování služeb je proces, který Erl (2009) dekomponuje do dvanácti kroků, které by měly být provedeny (některé z kroků jsou volitelné).

- 1) Dekompozice řídicího procesu – zdokumentovaný proces se rozloží na jednotlivé co nejatomičtější kroky. Může být využito vývojových diagramů, nebo nástrojů procesní analýzy.
- 2) Identifikace kandidátů operací řídicích služeb – některé činnosti procesu identifikované v předchozím kroku nemusí nutně patřit do logiky služby, a proto by ani neměly do kandidáta služby být zapouzdřeny. Jedná se především o manuální činnosti, které nelze automatizovat, nebo ani automatizovány být nemají.
- 3) Abstrakce instrumentační logiky – instrumentační logikou se myslí podniková pravidla, logika výjimek, logika podmínek a logika sekvencí. Instrumentační logika je opět součástí již dříve dekomponovanému procesu. Ve vývojovém diagramu se jedná o rozhodovací úseky typu: „Pokud je splněna podmínka, proveď akci x , pokud není splněna, proveď akci y .“

- 4) Vytvoření kandidátů řídicích služeb – v tomto kroku se jednotlivé činnosti procesu seskupí podle logického kontextu jejich činnosti. Z těchto kontextových skupin se již pomalu formují kandidáti služeb.
- 5) Aplikace principů servisní orientace – v tomto kroku jsou na doposud identifikované činnosti, které už jsou seskupeny podle kontextu, aplikovány dvě z vlastností služeb: znovupoužitelnost a autonomie. Činnosti mohou být slučovány, mohou být vytvořeny nové kontextové skupiny apod. tak, aby model vyhovoval podmínkám znovupoužitelnosti a autonomie.
- 6) Identifikace kandidátských kompozic služeb – podle scénářů, které by mohly při provádění modelovaných procesů nastat, se znovu vytvoří kompozice jejich kompozice. Smyslem této činnosti je poskytnout pohled na vhodné seskupování činností do kandidátů služeb, prozkoumat vztahy mezi vrstvami instrumentálních a řídicích služeb nebo také upozornit na jakoukoli možnou chybějící logiku.
- 7) Přezkum seskupení operací řídicích služeb – na základě předchozích kroků je vhodné přehodnotit doposud navrhnuté seskupení činností v jednotlivých skupinách.
- 8) Analýza aplikačních procesních požadavků – tento a následující kroky jsou volitelné a jsou vhodné hlavně pro velice rozsáhlá servisně orientovaná řešení. Smyslem je rozlišit řídicí a aplikační logiku a zanalyzovat, zda pro provedení určitých akcí v daném systému potřeba komunikace i s dalšími systémy.
- 9) Identifikace kandidátů operací aplikačních služeb – podobně jako v kroku 1) se provede i zde dekompozici procesů aplikační logiky. Nemělo by se stávat, že bude odkazováno na kroky řídicích procesů, pro které jsou funkce aplikační logiky identifikovány.
- 10) Vytvoření kandidátů aplikačních služeb – podobně jako v kroku 4) se provede seskupení podle kontextů, které následně formují kandidáty aplikačních služeb.
- 11) Přezkum kandidátů kompozic služeb – na kandidáty aplikačních služeb jsou aplikovány scénáře z kroku 5).
- 12) Přezkum seskupení operací aplikačních služeb – zde jsou opravena nevhodná seskupení a nedostatky, které byly odhaleny v předchozím kroku.

Servisně orientovaný návrh

Servisně orientovaný návrh je proces, během něhož se z logických kandidátů služeb vytvářejí konkrétní fyzické návrhy. Jako každý jiný proces se i tento skládá z několika kroků:

- sestavení SOA,
- návrh služeb.

Jednotlivé kroky budou blíže rozebrány v dalším textu.

Sestavení SOA

Sestavení servisně orientované architektury se dá označit za přípravnou fázi servisně orientovaného návrhu. Opět se jedná o proces skládající se ze tří kroků.

1. Vytvoření vrstev služeb

Prvním krokem servisně orientovaného návrhu je zamyšlení se nad vytvořením vrstev služeb. Aplikace rozvrstvení služeb v rámci navrhované SOA není povinné, ale pokud dojde k rozhodnutí o jejich vytvoření, mohou vzniknout tři typy vrstev.

Aplikační vrstva služeb obsahuje služby, které poskytují opětovně použitelné služby vycházející z obvyklých nástrojů a dovedností společnosti.

Vrstva služeb řízení se skládá ze služeb, jejichž funkční kontext je vyvozen ze vzorů řízení fungujících v podniku, které jsou následně přímo aplikovány na služby aplikační vrstvy. Pokud nejsou aplikační a řídicí vrstvy vytvořeny samostatně, pravděpodobně v systému existují služby hybridní, které obsahují jak aplikační, tak řídicí logiku.

Úkolem vrstvy služeb instrumentace je koordinovat spolupráci řídicích a aplikačních služeb. Služby obsažené v této vrstvě jsou odvozeny přímo z firemních procesů, proto se tato vrstva nazývá také jako vrstva rodičovských řídicích procesů.

2. Aplikace standardů

V tomto kroku jsou zvoleny standardy tak, aby co nejlépe podporovaly součásti navrhovaného servisně orientovaného řešení. Základním kamenem je využívání jazyka XML k reprezentaci dat a ve všech typech komunikace.

Společnost Web Service Interoperability Organization (WS-I) vydává standard WS-I Basic Profile, který se skládá ze sady specifikací neproprietárních webových služeb spolu s jejich upřesněními a změnami⁹. Jsou stanoveny struktury XML dokumentů používaných v rámci servisně orientovaných řešení

3. Volba rozšíření SOA

V tomto kroku se stanoví, které vlastnosti současné SOA bude navržené servisně orientované řešení splňovat. Plně závisí na tom, jaké jsou na navrhovaný systém požadavky, jelikož žádná z vlastností současné SOA nejsou povinné, návrhář si naopak může vybrat, které vlastnosti chce do svého systému zahrnout.

Návrh služeb

Cílem procesu návrhu služeb je vytvoření fyzického návrhu služeb, které odpovídají několika požadavkům, mezi něž patří omezení reálného světa kolem nás, podnikové požadavky, principy servisní orientace a zapouzdření takové množství logiky, jak je specifikováno v servisně orientované analýze.

Velice důležitou součástí servisně orientovaného návrhu, respektive návrhu služeb, jsou návrhové standardy, které mohou být využity už ve fázi servisně orientované analýzy, ale nemusí na ně být kladen tak velký důraz, neboť vytvoření kandidáti služeb mohou být dále modifikováni, kdežto jakmile je projekt dotažen do fáze návrhu, kdy jsou vytvářeny již fyzické podoby konkrétních služeb, stává se formalizace postupů na-prosto klíčovou.

V rámci této kapitoly je použito několik převzatých ukázek zdrojových kódů, které slouží pro ilustraci používaných technologií návrhu.

⁹ Zdroj: Ferris et al., 2007

Jazyk XSD

Jazyk XSD (XML Schema Definition Language) byl vytvořen, aby poskytoval formální hierarchickou strukturu XML dokumentů. S technologií webových služeb úzce souvisí, jak bude patrné v následujících částech.

Základním XSD elementem je `schema`, který obsahuje posloupnost atributů odkazujících na důležité obory názvů (tzv. namespace). Vždy je použit namespace `http://www.w3.org/2001/XMLSchema`, který vychází ze specifikace XSD.

Dalším důležitým XSD elementem je `targetNamespace`, který umožňuje přiřadit oboru názvu vlastní elementy vytvořené pomocí elementů `element`, `complexType` a `sequence`. XSD schémata je možné modularizovat, to znamená, že jedno XSD schéma může importovat obsah jiného. Toho je dosaženo pomocí elementů `import` a `include`.

Jazyk WSDL

Stěžejní technologií návrhu služeb je jazyk WSDL (Web Services Description Language), který byl jako standard vydán organizací W3C a současné době je dostupný ve verzi 2.0¹⁰. Jazyk WSDL je formátem jazyka XML a odpovídá na tři základní otázky, jak uvádí Weiss a Rychlý (2007):

- Jaké funkce poskytuje daná služba?
- Kde je daná služba uložena?
- Jak může být s danou službou navázána komunikace?

WSDL dokument, respektive popis služby, je rozčleněn do dvou elementárních částí, a to abstraktního a konkrétního popisu.

Úkolem abstraktního popisu je zavedení rozhraní služby, aniž by se specifikovalo, jaká technologie bude použita pro přenos zpráv mezi službami, což je velká výhoda,

¹⁰ Zdroj: Chinnici et al., 2007

neboť se do budoucna automaticky vyhýbáme problémům, které mohou nastat při změnách technologické platformy pro přenos zpráv. Abstraktní popis se skládá ze tří částí:

- `interface` – poskytuje abstraktní pohled na rozhraní služeb. V praxi to vypadá tak, že jsou vyjmenovány konkrétní operace, které daná služba poskytuje,
- `operation` – definuje jednotlivé operace v konstruktu `interface`, které se dále skládají z jednotlivých zpráv,
- `message` – definuje zprávy přijímané (`input`) a odesílané (`output`) jednotlivými operacemi.

Konkrétní popis pak spojuje logiku definovanou abstraktním popisem s konkrétní technologií přenosu, rovněž se skládá ze tří částí:

- `binding` – obsahuje popis požadavků na službu pro navázání konkrétních fyzických spojení, jinak řečeno obsahuje popis konkrétní technologie, která může být pro komunikaci použita (pravděpodobně nejběžnější technologií je protokol SOAP),
- `endpoint` – představuje konkrétní fyzickou adresu, na které bude daná služba dostupná,
- `service` – obsahuje skupinu elementů `endpoint`.

Základní struktura WSDL dokumentu popisujícího službu pak může vypadat následovně:

```

<definitions>

  <types>
    specifikace typů...
  </types>

  <message>
    specifikace zpráv...
  </message>

  <interface>
    specifikace rozhraní...
  </interface>

  <binding>
    specifikace binding....
  </binding>

</definitions>

```

Obrázek 3 - příklad dokumentu WSDL

Následují některé další hojně využívané elementy:

- `definitions` – jedná se o kořenový element všech WSDL dokumentů obsahující hlavně jmenné prostory, které jsou v rámci daného dokumentu používány,
- `types` – konstrukt, do něhož se umísťují XSD schémata a definice vlastních XSD elementů,
- `input` a `output` – při použití v konstruktu `operation` představují příchozí a odchozí typy zpráv, v konstruktu `binding` obsahují podrobnosti o komunikačních protokolech,

Jazyk SOAP

SOAP (pozn.: v minulosti zkratka pro Simple Object Access Protocol, dnes se již SOAP používá jako samostatný název) je komunikační protokol využívaný v rámci ser-

visně orientovaných řešení¹¹. V současné době je jeho specifikace W3C ve verzi 1.2¹². Jeho základní vlastnosti jsou nezávislost na transportních protokolech, bezstavovost a nezávislost na operačních systémech.

Zpráva, která dodržuje strukturu SOAP, prakticky vypadá jako obálka, v níž je vložen dopis, tak jako v klasickém světě kolem nás. Struktura SOAP zpráv je rozdělena do dvou částí, hlavičky (header) a těla zprávy (body).

Hlavička je nepovinná a skládá se z několika bloků, které v sobě obsahují metadata. Hlavička zpravidla obsahuje potřebné informace, pro všechny služby, které mohou danou zprávu obdržet. Tělo zahrnuje samotná data zprávy, která jsou zformátována v XML, dále sekci, která je věnována ošetření chybových stavů. SOAP rovněž umožňuje odesílání dat, která nesnadné nebo zcela nemožné popsat pomocí XML (binární soubory).

Základní XML elementy používané v SOAP dokumentu jsou `Envelope`, který je kořenovým elementem SOAP zprávy a obsahuje volitelný element `Header` a povinný element `Body`, přičemž element `Body` se definuje v rámci konstruktu `message` v WSDL dokumentu. Dalším nepovinným elementem je `Fault`, který se může vyskytnout v těle SOAP zprávy a jeho úkolem je poskytovat chybové hlášky v podobě srozumitelné pro člověka.

Shrnutí

V kapitole teoretická východiska jsem se při rozebírání problematiky servisně orientované architektury snažil postupovat systémově, tedy od celku k částem. Na začátku se zabývám historií a obecnými vlastnostmi SOA. Následuje představení základních komponent, ze kterých se obvykle servisně orientované řešení skládá. Dále jsou zmíněny vlastnosti nejpodstatnější části servisně orientovaných řešení. Další část kapito-

¹¹ Samozřejmě mohou být používány jakékoli jiné komunikační protokoly, ale SOAP je se SOA spjat v nejvyšší možné míře.

¹² Zdroj: Gudgin et al., 2007

ly pojednává o životním cyklu zavádění SOA, čímž se dostávám podrobnějšímu vysvětlení postupů servisně orientované analýzy a návrhu.

V rámci praktické části této práce budou použity výše popsané postupy, zejména servisně orientované analýzy a návrhu. U servisně orientovaného návrhu to pak budou hlavně značkovací jazyky založené na XML. Co se týče ostatních teoretických východisek zmíněných v této kapitole, jsou popsána proto, že jsou nedílnou součástí všech činností servisně orientované analýzy a návrhu.

Analýza a zhodnocení současného stavu architektury IS podniku

V této kapitole bude nastíněn současný stav části informačního systému společnosti ON Semiconductor a bude nastíněn problém, jehož vyřešení je jedním z cílů této diplomové práce.

Prvním cílem této práce bylo vysvětlení pojmu servisně orientovaná architektura, druhým cílem bude aplikace této teorie v části informačního systému společnosti ON Semiconductor.

Rozhovor se zaměstnanci

Podkladem pro analýzu současného stavu subsystému informačního systému společnosti ON Semiconductor je rozhovor se zaměstnancem ERDC v Rožnově pod Radhoštěm. Struktura rozhovoru je zobrazena v tabulce 2. Je použita metodika, kterou popisuje Kaluža (2010).

Otázky pro rozhovor jsem volil tak, aby mi poskytly obraz fungování subsystému informačního systému. Zaměřil jsem se zejména na oblast ukládání dat plynoucích z výroby a dále pak na záměr společnosti ON Semiconductor na další rozvoj v této oblasti. Hlavním smyslem rozhovoru pak byla identifikace problémové oblasti v subsystému.

Tabulka 2 - Rozhovor se zaměstnancem

Respondent: Ing. Petr Kahánek, Ing. Radek Fojtík	Tazatel: Bc. Michal Hanzal
Místo: ERDC Rožnov pod Radhoštěm	
Cíl: Popis stávajícího subsystému a identifikace požadavků na jeho změnu.	Podklady: Mnoholetá praxe respondenta a jeho znalost podnikových systémů ON Semiconductor.

<p>Program:</p> <ul style="list-style-type: none"> • Úvod • Podstata problému • Struktura rozhovoru • Přehled témat rozhovoru • Shrnutí • Dotazy respondenta • Závěr 	<p>Časový odhad:</p> <p>cca 60 minut</p>
<p>Otázka č. 1:</p> <p>Jak jsou v podniku uchovávána data z výroby?</p>	<p>Odpověď:</p> <p>V současnosti jsou v provozu dvě rozsáhlé databáze – Synergy a dataPOWER™. V obou jsou ukládána identická data s tím, že databáze dataPOWER™ je součástí komplexnějšího podnikového řešení, které jsme nakoupili od společnosti PDF Solutions. Databáze Synergy vznikla v minulosti a do doby nákupu řešení dataPOWER™ plnila funkci hlavní databáze na uchování dat plynoucích z výroby.</p>
<p>Otázka č. 2:</p> <p>Proč jsou data doposud ukládána do dvou nezávislých databází?</p>	<p>Odpověď:</p> <p>Databáze dataPOWER™ je primárně součástí komplexnějšího analytického řešení. Databáze Synergy je v současnosti v provozu hlavně z toho důvodu, že poskytuje data našim vlastním analytickým aplikacím.</p>
<p>Otázka č. 3:</p>	<p>Odpověď:</p>

<p>Které aplikace tedy s databází Synergy pracují?</p>	<p>V současnosti jsou data za Synergy využívány třemi aplikacemi:</p> <ul style="list-style-type: none"> • Vlastní webové rozhraní pro tvorbu dotazů do databáze Synergy pro uživatele bez znalostí SQL dotazovacího jazyka, • CDEP, • SPC.
<p>Otázka č. 4:</p> <p>Kdo a k čemu využívá aplikaci CDEP?</p>	<p>Odpověď:</p> <p>Aplikace CDEP je využívána zaměstnanci design centra, které se zabývá návrhem a vývojem nových technologií, které se plánují v budoucnu v ON Semiconductor vyrábět.¹³</p>
<p>Otázka č. 5:</p> <p>Kdo a k čemu využívá aplikaci SPC?</p>	<p>Odpověď:</p> <p>Aplikace SPC je využívána device engineers, kteří mají za úkol sledovat proces výroby a případně korigovat nastavení výrobních zařízení tak, aby nedocházelo k odchylkám od stanovené kvality výrobků.¹⁴</p>
<p>Otázka č. 6:</p> <p>Proč jste se rozhodli pro nákup řešení da-</p>	<p>Odpověď:</p> <p>Důvodů k tomuto rozhodnutí bylo několik.</p>

¹³ Bližší popis CDEP v podkapitole Jednotlivé části řešeného subsystému.

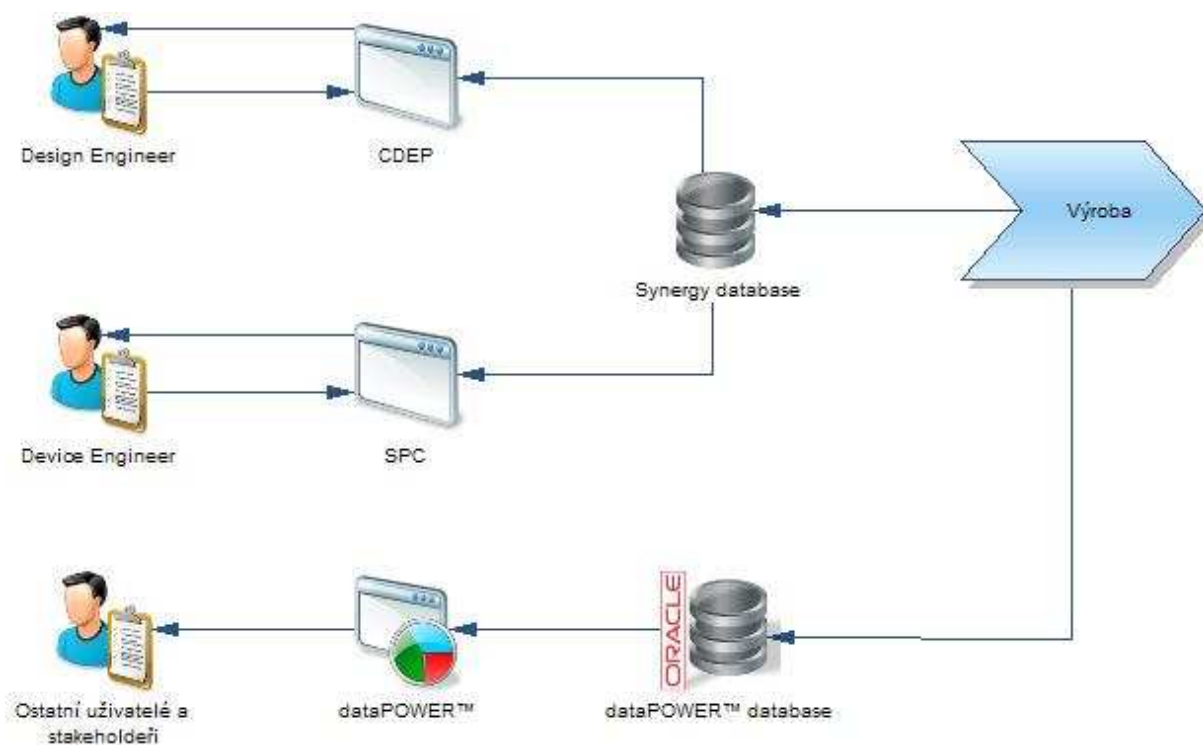
¹⁴ Bližší popis SPC v podkapitole Jednotlivé části řešeného subsystému.

taPOWER™?	Jedním z hlavních pak bylo to, že dataPOWER™ je určeno speciálně pro společnosti, které se zabývají výrobou polovodičových součástek. V databázi je téměř dokonalý strukturní pořádek. Společnost PDF Solutions navíc poskytuje vynikající servis a podporu.
<p>Otázka č. 7:</p> <p>Jak hodláte řešit dvojí ukládání výrobních dat?</p>	<p>Odpověď:</p> <p>Do budoucna hodláme plně přejít k využívání databáze dataPOWER™ i pro aplikace CDEP a SPC. V rámci své stáže jste pro nás již vytvořil procedury, které nám dovolují vybírat požadovaná data využívaná v CDEP a SPC. Dále v rámci vaší diplomové práce bychom byli rádi, abyste se pokusil využít servisně orientovaných přístupů a navrhl rozhraní pro přístup k těmto datům.</p>
<p>Otázka č. 8:</p> <p>Proč hodláte využít principů SOA?</p>	<p>Odpověď:</p> <p>Jedná se o poměrně nový přístup k návrhu informačních systémů a myslíme si, že orientace na SOA může mít do budoucna pro společnost strategický význam.</p>

Z rozhovoru provedeného se zaměstnanci ERDC jasně vyplývá, že v systému jsou stejná data ukládána dvakrát, což je v současné době nezbytné z důvodu existence dvou sad aplikací, které používají data ze dvou databází. V budoucnu se ovšem plánuje přechod k využití jen jedné z databází. Návrh řešení je součástí této diplomové práce v podobě jejího cíle.

Jednotlivé části řešeného subsystému

Část informačního systému, na které bude provedena aplikace teoretických poznatků servisně orientovaných architektur, se skládá několika prvků, jak vyplynulo z rozhovoru se zaměstnanci ERDC (obrázek 8). V následujícím textu budou jednotlivé složky popsány.



Obrázek 4 - Schéma části informačního systému ON Semiconductor

Proces výroby čipů a polovodičových součástek generuje obrovské množství dat, která jsou ukládána do dvou datových úložišť, databází Synergy a dataPOWER™. Databáze Synergy byla vybudována v minulosti, aby plnila funkci hlavního datového skladu pro uchování dat plynoucích z výroby. Později tyto funkce převzala databáze dataPOWER™ a Synergy je v dnešní době používána jen za účelem provozu aplikací CDEP a SPC, které budou popsány později. To znamená, že některá data jsou ukládána dvakrát, jednou do databáze dataPOWER™ a podruhé do databáze Synergy v každé ovšem jiným způsobem.

Core Device Electrical Parameters (CDEP)¹⁵

Jedná se o aplikaci, která slouží návrhovým inženýrům. Umožňuje vybírat historická data uložená v Synergy a na nich následně simulovat chování nově navrhovaných součástek. Design inženýři mají možnost sledovat celou řadu statistických dat a parametrů, které jsou stanoveny podle technologického postupu výroby jednotlivých součástek. U nově navrhovaných výrobních technologií se nastavují parametry, které se budou na vyráběných součástkách měřit a jsou stanoveny množiny limit, pro stanovení, kdy je součástka funkční a kdy se jedná o zmetek. Jakmile je součástka nebo výrobní technologie v CDEP navržena a odpovědnými osobami schválena, je automaticky převedena do aplikace SPC, která je popsána v následujícím textu.

Statistic Process Control (SPC)¹⁶

Jedná se o komplexní aplikaci, která je navržena pro proaktivní sledování parametrů statistických hodnot důležitých parametrů polovodičových součástek pro potřeby tzv. device inženýrů, jejichž úkolem je zabezpečovat správný průběh výroby a tím zajistit minimalizaci produkce zmetků. Hlavními funkcemi aplikace SPC je statistické sledování výsledků měření vyrobených součástek, které se provádí mezi jednotlivými fázemi výrobního procesu. Při překročení stanovených limit jsou device inženýři na tuto skutečnost upozorněni a překontrolují a upraví výrobní proces. Do databáze jsou z výroby ukládána raw data a pro potřeby aplikace SPC jsou zpracovávána speciálním deamonem, který nefunguje reálnodobě, ale každá výrobní jednotka si nastavuje časový úsek, pro který mají být data zpracována a zobrazena aplikací SPC.

Obě dvě výše uvedené aplikace, CDEP i SPC, jsou produktem, který byl pro pro ON Semiconductor vyvinut a je spravován v European Region Development Center.

¹⁵ Zdroj: ERDC – osobní konzultace, předvedení aplikace zaměstnancem, CDEP User's Manual

¹⁶ Zdroj: ERDC – osobní konzultace, předvedení aplikace zaměstnancem, SPC Admin Guide

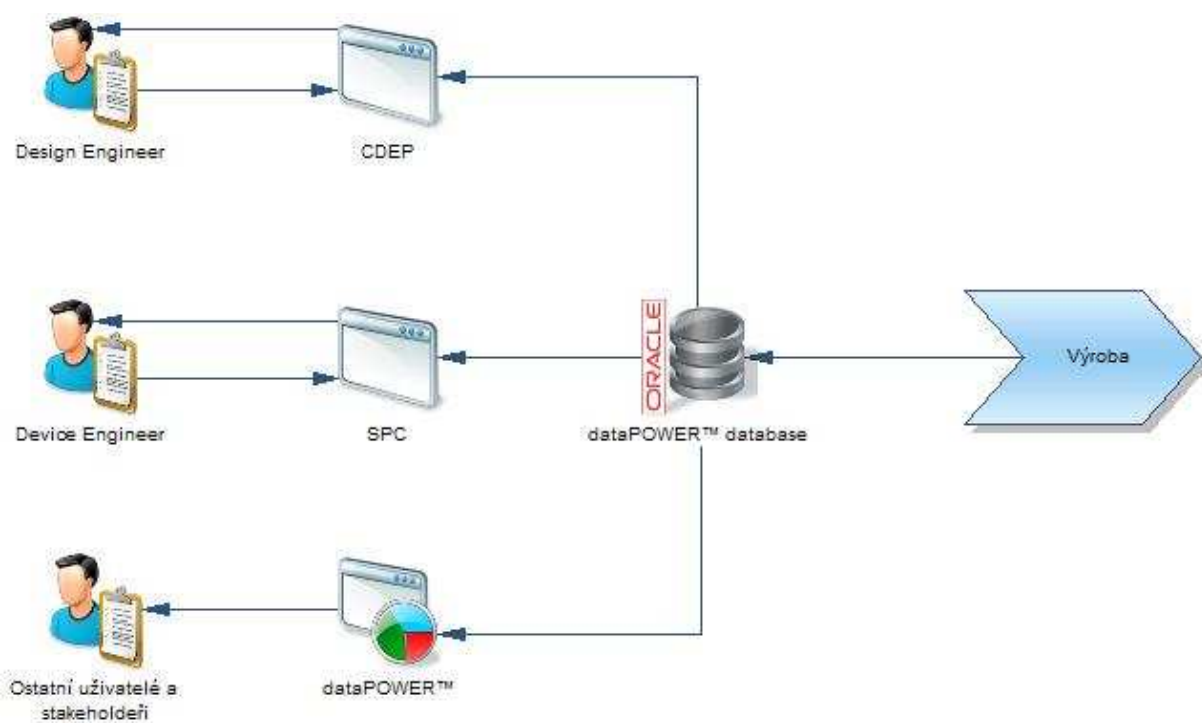
dataPOWER™¹⁷

Řešení dataPOWER™ je soubor vysoce výkonných aplikací a databáze od společnosti PDF Solutions. Jedná se o řešení, navržené speciálně pro závody, které se zabývají výrobou polovodičových součástek, a poskytuje tedy nejrozličnější statistické funkce pro sledování výroby a odstraňování závad. Hlavní součástí, na níž je celé řešení postaveno, je vysoce výkonná databáze, v případě společnosti ON Semiconductor se jedná o databázový stroj Oracle 10g.

Konečný požadovaný stav subsystému informačního systému

Vzhledem k tomu, že dochází k ukládání duplikátních dat do dvou nezávislých databází, je strategií společnosti ON Semiconductor do budoucna postupně vyřadit z používání zastaralou databázi Synergy a plně přejít k maximálnímu využití databáze dataPOWER™, která podle názoru zaměstnanců ERDC lépe strukturovaná, spolehlivější a je k ní poskytována mimořádně kvalitní dokumentace a podpora. Požadovaný stav subsystému, který vyplývá z požadavků rozhovoru provedeného se zaměstnanci společnosti ON Semiconductor (tabulka 2), je zobrazen na obrázku 9. Úskalí ke kompletnímu přechodu k užívání výhradně databáze dataPOWER™ je v tom, že data jsou uložena jiným způsobem, takže bude nezbytné provádět určité transformace, než budou SPC a CDEP schopny s daty z databáze dataPOWER™ pracovat.

¹⁷ Zdroj: ERDC – osobní konzultace, zkušenosti získané během stáže, PDF Solutions, 2008



Obrázek 5 - Konečný požadovaný stav subsystému informačního systému

Návrh architektury podle servisně orientovaného přístupu

Jak už bylo nastíněno v rámci kapitoly Analýza a zhodnocení současného stavu architektury IS podniku, druhým cílem této diplomové je návrh přebudování architektury subsystému za využití servisně orientovaných přístupů v podobě rozhraní, které bude sloužit aplikacím SPC a CDEP k přístupu k datům v datovém skladu dataPOWER™.

Tato kapitola se skládá ze dvou částí:

- servisně orientované analýzy, která stanoví rozsah řešení problému, a jejím výstupem budou kandidáti služeb.
- servisně orientovaného návrhu, kde budou navrženy konkrétní služby.

V této a další části s názvem Servisně orientovaný návrh jsem postupoval v souladu s metodikou, kterou jsem popsal v teoretické části této práce.

Servisně orientovaná analýza

Definice rozsahu analýzy

V rámci servisně orientované analýzy jsem, jak už vyplývá z požadovaného řešení definovaného problému, zanalyzoval proces komunikace mezi aplikacemi CDEP a SPC s databází Synergy, aby po té mohl být navrženo rozhraní pro načítání dat z dataPOWER™ databáze.

Servisně orientovaná analýza bude zaměřena na část systému, kde dochází ke komunikaci mezi aplikacemi CDEP a SPC s databází Synergy a poté možné napojení na databázi dataPOWER™.

Tato práce vznikla za podpory společnosti ON Semiconductor na základě absolvování studentské stáže v ERDC v Rožnově po Radhoštěm. V rámci této stáže byl autorovi poskytnut přístup k některým informačním systémům společnosti ON Semiconductor a k řešení zadán návrh a vývoj rozhraní pro přístup k datům v datovém skladu.

Cílem této práce je navrhnout servisně orientované řešení, které by se pohybovalo v mezích zadání návrhu a vývoje rozhraní pro přístup k datům a využilo již v rámci studentské praxe vytvořené řešení.

Identifikace stávajících systémů

Jak už bylo zmíněno výše, stávající systémy, které budou v této práci hrát určitou roli, jsou uživatelské aplikaci CDEP a SPC, dále pak databáze Synergy a dataPOWER™.

V rámci dlouhodobého plánu společnosti ON Semiconductor přechodu k maximálnímu využití databáze dataPOWER™ byla v rámci studentské stáže autora této práce vytvořena sada procedur, které vybírají a transformují data, tak aby byla připravena pro použití pro aplikace CDEP a SPC.

Popis databáze Synergy

Na obrázku 10 je vyobrazeno schéma databáze Synergy, z níž jsou v současné době extrahována data pro aplikace CDEP a Synergy. Schéma jsem překreslil ze schématu, které mi bylo poskytnuto zaměstnanci ERDC a které jsem využíval po dobu své stáže. Na tomto místě je vhodné vysvětlit některé pojmy¹⁸ využívané v průmyslovém odvětví výroby polovodičových součástek, integrovaných obvodů a čipů.

Wafer (deska) – je velice tenká destička ve tvaru kruhu, která vznikne rozřezáním uměle vytvořeného monokrystalu křemíku o extrémní čistosti dosahující téměř 100% a minimálním výskytu kazů v krystalické mřížce. Wafery jsou používány v různých velikostech od jednoho do 18 palců při tloušťkách od 275 do 925 μm . V případě závodu společnosti ON Semiconductor v Rožnově pod Radhoštěm jsou využívány wafery o průměru 4 a 6 palců a tloušťkách 525 a 675 μm (wafery jsou rozřezány z křemíkového monokrystalu, který společnost sama vyrábí), na které jsou různými specifickými technologiemi nanášeny další vrstvy, jež následně tvoří jednotlivé součástky. Na jeden wafer

¹⁸ Zdroj: konzultace se zaměstnanci ON Semiconductor

se směstná několik desítek až tisíců čipů v závislosti na průměru desky a velikosti vyráběné součástky.

Lot (sada) – jedná se o sadu jednotlivých waferů, které prochází celým proces společně v jednom souboru. Vždy se jedná o wafery, na kterých se vyrábí stejný typ součástky stejným technologickým postupem.

LOTEVENTS

Tabulka LOTEVENTS eviduje informace o tom, co se daným lotem dělo. Konkrétně je zaznamenáno, kterého lotu se záznam týká, datum události, která je zaznamenána, jméno a verze testovacího programu, výrobní technologie lotu, výrobní proces a další.

LOTDATA

Tabulka LOTDATA obsahuje výsledky testů, které byly na daném lotu provedeny v několika statistických ukazatelích, jako jsou minimum a maximum, průměr, standardní odchylka, medián apod. V podobě cizích klíčů obsahuje vazby na tabulky LOTEVENTS a PARAMETERS.

PARAMETERS

Tabulka PARAMETERS obsahuje informace o různých parametrech testů, které jsou na vyráběných součástkách měřeny. V tabulce se pro každý evidovaný parametr ukládá jeho název, skupina, do které patří, jednotky a různé limity, které indikují, zda se jedná o vdanou součástku nebo ne.

WAFERPATH

Tabulka WAFERPATH obsahuje informace o tom, které wafery patří do daného lotu.

WAFERDATA

Tabulka WAFERDATA obsahuje podobná data jako tabulka LOTDATA, ovšem v tomto případě se týkají každého waferu zvlášť. Je to dáno tím, že některé testy se pro-

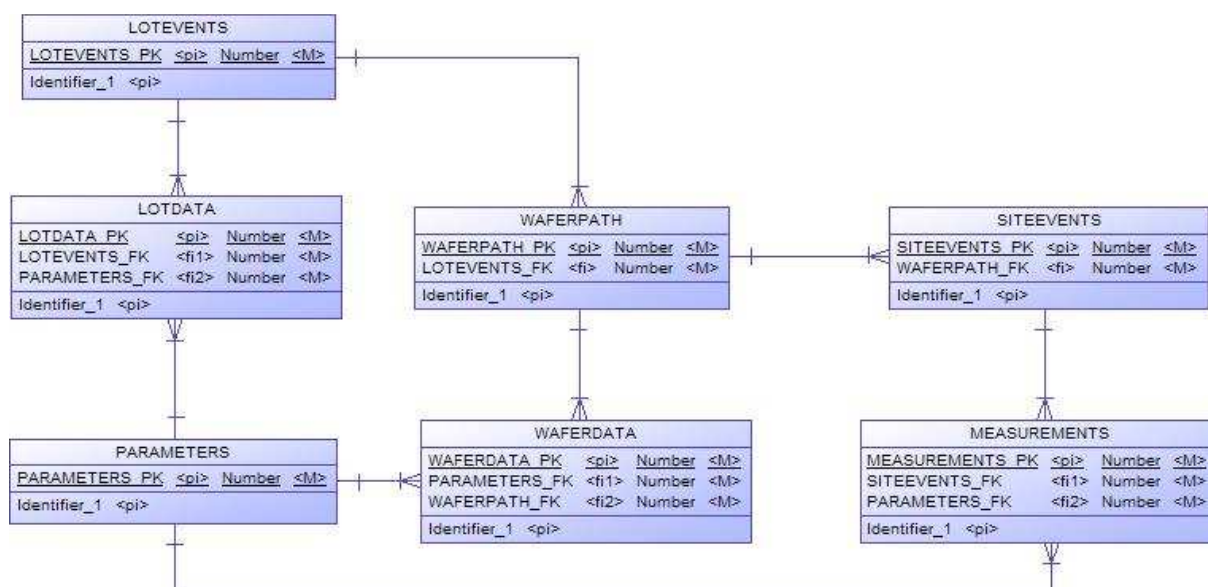
vádějí výběrem pouze několika vzorových desek pro celý lot (takovéto testy probíhají na speciálních testovacích oblastech, jejichž umístění je na každém waferu identické), jiné testují každou vyrobenou součástku na waferu zvlášť.

SITEEVENTS

Tabulka SITEEVENTS obsahuje názvy a souřadnice testovacích oblastí na waferech.

MEASUREMENTS

Tabulka MEASUREMENTS přiřazuje k jednotlivým souřadnicím na waferu sledované parametry.



Obrázek 6 - Schéma databáze Synergy (Zdroj: interní materiály ON Semiconductor)

Popis databáze dataPOWER™

Databáze dataPOWER™ je součástí komplexního řešení pro podniky vyrábějící polovodičové součástky od společnosti PDF Solutions. Jsou v ní ukládána stejná data, jako v databázi Synergy, ovšem zcela jiným způsobem. V další části textu bude popsána jen její část, která je pro tuto práci podstatná, jelikož jako celek obsahuje několik stovek

tabulek a pohledů, které v kontextu této práce nejsou důležité a je možno od nich abstrahovat. Schéma části databáze dataPOWER™ je zobrazeno na obrázku 11¹⁹.

OP_LOG

Tabulka OP_LOG obsahuje podobná data jako tabulka LOTEVENTS v databázi Synergy, tzn., že pro každý lot, který projde kontrolním měřícím procesem, se zde objeví záznam.

LOT

Tabulka LOT slouží k detailnímu popisu jednotlivých lotů. Velice důležitým údajem je lot_id, což je identifikační kód každého vyrobeného lotu, jako cizí klíč je využit v mnoha tabulkách, mimo jiné také v tabulce OP_LOG jako identifikace zpracovaného lotu a identifikace tzv. „parent“ lotu (rodičovského lotu – v průběhu výroby může dojít k situaci, kdy se jeden lot rozdělí do dvou nebo více dalších lotů).

WAFER

Tabulka WAFER obsahuje data, kterými je detailně popsán každý wafer.

PROGRAM

Tabulka PROGRAM obsahuje data, která detailně popisují, testovací procedury, které jsou ve výrobě na letech prováděny. Velice důležité jsou atributy:

- pg_key – automaticky generovaný primární klíč tabulky v číselném formátu; využívá se při konstrukci názvů tabulek, které obsahují surová data z testovacích měření (princip bude vysvětlen dále).
- pgc_key – identifikuje speciální seskupení programů podle typů měření.

PROG_CLASS

¹⁹ Zdroj popisu a z něj odvozeného obrázku 11 je DataPOWER Data Dictionary, 2007 a zkušenosti nabyté během stáže

Tabulka PROG_CLASS obsahuje definice speciálních skupin testovacích testovacích procedur.

PRODUCT

Tabulka PRODUCT slouží k popisu jednotlivých výrobků, které jsou ve výrobní jednotce produkovány. Mimo jiné obsahuje název produkty a informace o pracovnících odpovědných za daný produkt.

TECHNOLOGY

Tabulka TECHNOLOGY obsahuje informace o dané výrobní technologii, která je použita při výrobě jednotlivých produktů.

FAMILY

Tabulka FAMILY popisuje dostupné skupiny, respektive rodiny, do nichž můžou jednotlivé produkty patřit.

PROG2LOT, PROD2LOT, PROD2FAM, PROG2TECH

Tabulky, které vznikly ze vztahů o kardinalitě M:N mezi tabulkami LOT, PROGRAM, PRODUCT, FAMILY a TECHNOLOGY

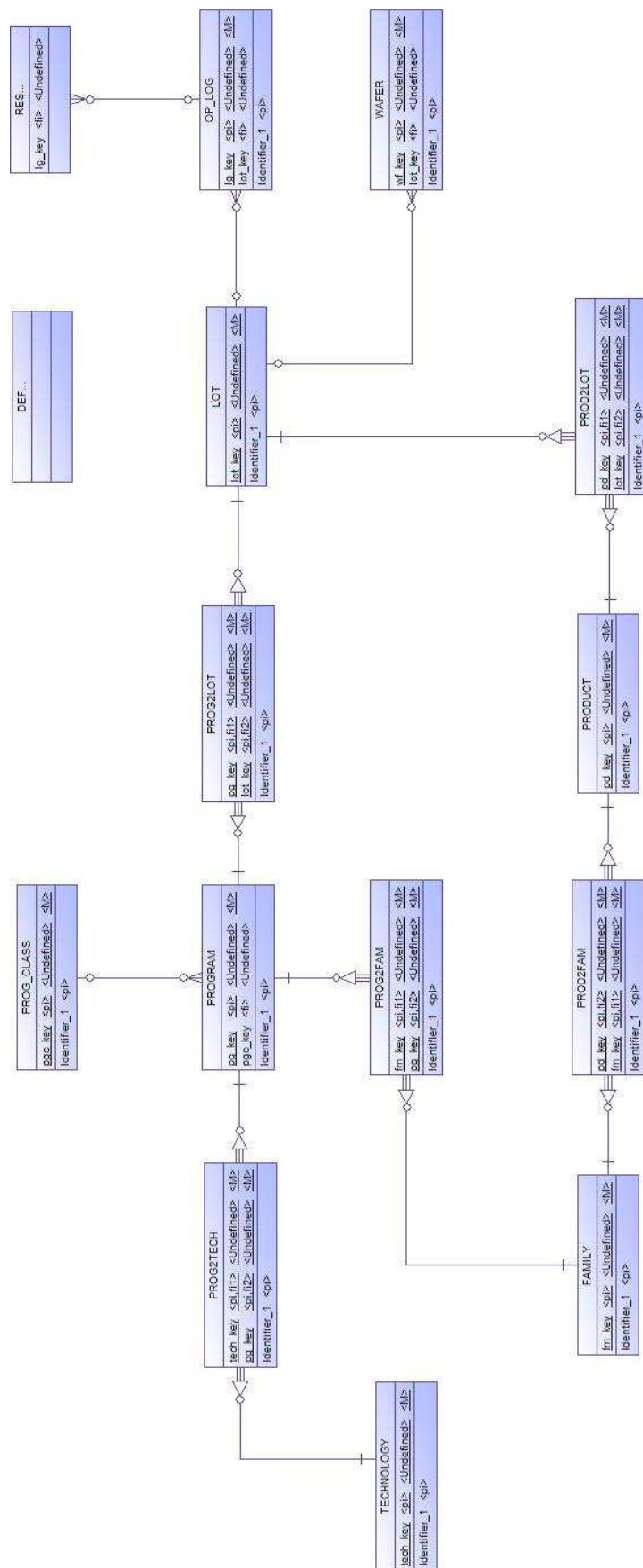
RES...

Soubor tabulek, které obsahují data s výsledky měření za jednotlivé loty. Jedná se soubor tabulek, které jsou generovány automaticky s tím, jak se postupně přidávají testovací programy. Jejich názvy jsou zkonstruovány z předpony RES („results“) a čísla daného testovacího programu (např.: RES1234 – „výsledková“ tabulka pro program 1234). Každá tabulka pak obsahuje výsledky za jeden testovací program pro všechny měřené loty.

DEF...

Opět se jedná o soubor tabulek, jejichž název je konstruován podobně, jako je tomu v případě tabulek RES... Tabulky DEF... obsahují detailní popis daného testovacího programu včetně různých stanovených limit a měrných jednotek. Specifikum této tabulky je to, že neobsahuje žádný primární ani cizí klíč, jediným spojením se strukturou tabulek v databázi dataPOWER™ je tedy číslo testovacího programu v názvu jednotlivých tabulek.

Smyslem tohoto detailního popisu obou databází je poukázání na rozdílný způsob uložení dat v jednotlivých databázích a tedy potřeba jejich transformace pro použití aplikacemi CDEP a SPC. Jak jsem uvedl již dříve, v rámci stáže v ON Semiconductor jsem vytvořil procedury, které tyto výběry a transformace provádějí a jejich popis je součástí této práce v podobě přílohy.



Obrázek 7 – Schéma části databáze dataPOWER™

Modelování kandidátských služeb

Při modelování kandidátských služeb jsem postupoval v souladu s metodami a postupy, které jsem popsal v kapitole Teoretická východiska návrhu servisně orientované architektury v podkapitole Servisně orientovaná analýza.

Smyslem a výsledkem modelování kandidátských služeb a návrhu služeb je WSDL specifikace služeb. WSDL dokument se pak stává konečným produktem servisně orientované analýzy a návrhu.

Pro návrh kandidátských služeb je klíčový proces výběru a transformace požadovaných dat z databáze dataPOWER™, jehož kroky jsem prozkoumal v rámci své stáže. Proces začíná tím, že uživatel aplikace CDEP nebo SPC dá podnět k extrakci dat z datového skladu. Aplikace podle uživatelem specifikovaných parametrů, což jsou v našem případě číslo lotu, vlastník databáze a program class key, odešle požadavek na data databázi. Žádost o data v databázi spustí proceduru, která data vybere a připraví ve formě pole na odeslání zpět aplikacím. Data jsou následně z databáze odeslána aplikacím k prezentaci (proces je zobrazen na obrázku 12).



Obrázek 8 – Proces výběru dat

1. Dekompozice procesu

Proces výběru dat jsem podle metodiky modelování kandidátských služeb dekomponoval na jednotlivé atomické kroky:

- uživatel požaduje data z databáze,
- uživatel specifikuje vstupní parametry,
- aplikace odesílá požadavek databázi,
- předání vstupních parametrů databázi,
- databáze přijímá vstupní parametry,
- spouští se procedura na výběr dat,
- příprava pole s požadovanými daty,
- odeslání dat aplikaci,
- data jsou přijata aplikací,
- data jsou prezentována uživateli.

2. Identifikace kandidátů operací služeb

Další krok metodiky požaduje zvolit kandidáty operací služeb, které bude služba zapouzdřovat, respektive vykonávat, což znamená, že jsem každý jednotlivý krok posoudil podle toho, kdo nebo co jej vykonává. Kroky, které nemohou být službou zapouzdřeny, jsem následně vyloučil. Jedná se o takové činnosti, které jsou vykonávány člověkem a nemohou být automatizovány v podobě logiky operace služby, nebo takové, které již vykonává jiná aplikace a není vhodné ani žádoucí je službě přiřadit. Vyloučené i službě přiřazené operace jsou v následujícím seznamu, jenž vychází ze předchozího seznamu nekomponujícího proces výběru dat na jednotlivé kroky:

- ~~uživatel požaduje data z databáze~~ – manuální krok prováděný uživatelem,
- ~~uživatel specifikuje vstupní parametry~~ – manuální krok prováděný uživatelem,
- ~~aplikace odesílá požadavek databázi~~ – provádí aplikace, se kterou pracuje uživatel,
- předání vstupních parametrů databázi,

- ~~databáze přijímá vstupní parametry~~ – vykonává databáze,
- ~~spouští se procedura na výběr dat~~ – vykonává extrakční procedura,
- ~~příprava pole s požadovanými daty~~ – vykonává extrakční procedura,
- odeslání dat aplikaci,
- ~~data jsou přijata aplikací~~ – vykonává uživatelská aplikace,
- ~~data jsou prezentována uživateli~~ – prezentuje uživatelská aplikace.

Identifikoval jsem tedy dvě operace, které bude služba vykonávat. Jedná se o předání vstupních parametrů specifikovaných uživatelem databázi a odeslání databází vybraných dat zpět aplikaci k dalšímu zpracování nebo prezentaci uživateli.

3. Abstrakce instrumentační logiky

V rámci výše popsaného procesu se nevyskytují žádné rozhodovací sekvence a ani nebyl vznesen požadavek na vytvoření vrstvy instrumentace služeb.

Vrstva instrumentace služeb jsou vytvářeny v rozsáhlých mnohvrstevných architekturách, kde plní funkci řízení spolupráce jednotlivých služeb. V rámci této diplomové práce nenavrhují takto rozsáhlé prostředí, proto jsem se ani nezabýval problematikou tvorby instrumentační vrstvy.

4. Vytvoření kandidátů služeb

V kroku číslo 2. Identifikace kandidátů operací služeb jsem identifikoval dvě operace, které by měla navrhované služby provádět:

- předání vstupních parametrů databázi,
- zpracování a odeslání dat aplikaci.

Nyní je nutné se rozhodnout, kolik služeb bude jednotlivé operace zapouzdřovat. Podle kontextu činností se jeví, že by mohly být seskupeny do jedné služby, která by vykonávala dvě operace.

5. Aplikace principů servisní orientace

V tomto kroku jsou podstatné principy znovupoužitelnosti a autonomie. Lze konstatovat, že služba je autonomní, jelikož má kontrolu nad veškerým zpracováním, které

má provádět. Dále lze konstatovat, že služba je znovupoužitelná, protože již teď je navrhována pro použití dvěma relativně nezávislými aplikacemi a je připravena být využita i dalšími aplikacemi, které by přistupovaly k databázi dataPOWER™ v podobném kontextu.

Následné kroky (6. až 12. – viz kapitola Teoretická východiska návrhu servisně orientované architektury, Servisně orientovaná analýza) jsou pro zpracování v této situaci zbytečné, neboť se týkají rozsáhlých, mnohovrstevných architektur, které jsou složeny z řady různých služeb. V kontextu této práce tedy byla identifikována jedna služba, jejímž úkolem bude komunikace s databází dataPOWER™. Návrh této služby dále popisují v další části této kapitoly.

Servisně orientovaný návrh

V této části diplomové jsem provedl servisně orientovaný návrh, kdy jsem vycházel z kandidáta služby, jenž byl identifikován v rámci servisně orientované analýzy v předchozí kapitole. Opět jsem vycházel z metodiky, která je popsána v teoretické části této práce, která se týká servisně orientovaného návrhu.

Sestavení SOA

Prvním krokem v procesu sestavení SOA je vytvoření vrstev služeb, které jsou rozděleny na tři typy. Jak vyplynulo ze servisně orientované analýzy, bude navržena jedna služba, která bude plnit funkci rozhraní pro přístup k databázi. To znamená, že rozvrstvení služeb v rámci této práce provedeno nebude. Pokud by šlo o zařazení služby do jedné ze tří kategorií vrstev služeb, jednalo by se pravděpodobně o službu aplikační vrstvy.

Aplikace standardů

V rámci návrhu služby jsem uplatnil následující standardy:

- Extensible Markup Language (XML) Version 1.0
- XML Schema
- WS-I Basic Profile Version 2.0
- Web Services Description Language (WSDL) Version 2.0

Návrh služby

Výsledkem servisně orientovaného návrhu je WSDL specifikace služby, z něhož se následně vychází při procesu vývoje služeb. Na obrázku 13 je konečná fyzická podoba služby zapsaná jazykem WSDL. V další části textu popisují jednotlivé úseky kódu.

Celý WSDL dokument začíná deklarací verze XML, která je v dokumentu použita: `<?xml version="1.0" ?>`. Následuje `<definitions ... >`, kde je specifikován název (byl zvolen název Komunikace) služby a takzvané jmenné prostory (name-

space), které obsahují URI umístěný daných dokumentů v síti²⁰ - včetně odkazů na standardy, které jsou vydány organizacemi jako w3c.

Oddíl `<types> ... </types>` je vlastně částí, která by mohla být uložena v samostatném XSD dokumentu, ale je možné jí integrovat do dokumentu WSDL, tak jak jsem to provedl v tomto případě. Oddíl `<types> ... </types>` definuje typy, se kterými bude služba pracovat. V tomto případě jsem definoval dva složené typy:

- VstupniParametry – jedná se o parametry, které jsou specifikovány uživatelem a odeslány službě Komunikace pro zpracování databází. Mezi tyto parametry patří LotID (číslo lotu, pro který jsou požadována data z databáze dataPOWER™), Owner (vlastník, respektive jméno databázového schématu, z něhož se mají data vybrat) a PGCKey (tzv. program class key, neboli číslo skupiny programů, jejíž výsledky měření uživatele zajímají).
- Vystup – jedná se o atributy výstupních dat. Patří mezi ně WaferKey (čísla jednotlivých waferů, která patří do zadaného lotu), CoordX (souřadnice x měřeného bodu, respektive součástky, na daném waferu), CoordY (souřadnice y měřeného bodu, respektive součástky, na daném waferu), Value (hodnota změřené na součástce na specifikovaných souřadnicích měřícím programem), TestName (jméno testu, který měření provedl) a TestUnits (jednotky provedeného testu).

V konstruktech `<message> ... </message>` specifikuji, jaké zprávy bude daná služba zpracovávat nebo odesílat a které typy bude používat daná zpráva. V tomto případě je to zpráva VstupniParametryMsg, která bude pracovat s typem VstupniParametry, a zpráva DataMsg, která pracuje s typem Vystup.

Nakonec je v abstraktní části WSDL definice služby specifikováno rozhraní služby, které zprávy rozlišuje na vstupní a výstupní. V případě služby Komunikace je vstupní zprávou VstupniParametryMsg a výstupní zpráva DataMsg.

²⁰ V tomto případě jsem jmenné prostory specifikoval jen obecně.

```

<?xml version="1.0"?>
<definitions name="Komunikace"

targetNamespace="http://cilovyNS.cz/komunikace.wsdl"
  xmlns:tns="http://cilovyNS.cz/komunikace.wsdl"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns="http://schemas.xmlsoap.org/wsdl/">

  <types>
    <schema targetNamespace="http://cilovyNS.cz/"
      xmlns="http://www.w3.org/2000/10/XMLSchema">
      <element name="VstupniParametry">
        <complexType>
          <sequence>
            <element name="LotID" type="string"/>
            <element name="Owner" type="string"/>
            <element name="PGCKey" type="integer"/>
          </sequence>
        </complexType>
      </element>
      <element name="Vystup">
        <complexType>
          <sequence>
            <element name="WaferKey" type="integer"/>
            <element name="CoordX" type="integer"/>
            <element name="CoordY" type="integer"/>
            <element name="Value" type="float"/>
            <element name="TestName" type="string"/>
            <element name="TestUnits" type="string"/>
          </sequence>
        </complexType>
      </element>
    </schema>
  </types>

  <message name="VstupniParametryMsg">
    <part name="vstup" element="VstupniParametry"/>
  </message>

  <message name="DataMsg">
    <part name="vystup" element="Vystup"/>
  </message>

  <interface name="Komunikace">
    <operation name="PrenosDat">
      <input message="tns:VstupniParametryMsg"/>
      <output message="tns:DataMsg"/>
    </operation>
  </interface>

  ...

</definitions>

```

Obrázek 9 - WSDL definice vytvořené služby

Zhodnocení výsledků zpracovaného řešení

V rámci předchozí kapitoly jsem provedl servisně orientovanou analýzu a návrh, jejichž výsledkem je WSDL specifikace služby Komunikace. Pokud se pozorně podíváme na životní cyklus servisně orientovaných řešení, servisně orientovaná analýza a návrh nejsou konečnými fázemi vývoje servisně orientované architektury – následují procesy vývoje, testování a zavádění služeb.

WSDL dokument služby, který jsem v rámci této práce vytvořil, bude předán ERDC ON Semiconductor k provedení dalších kroků podle životního cyklu.

Ve fázi vývoje služeb dochází k samotné konstrukci fyzické podoby služeb. Úkolem zaměstnanců ERDC tedy bude zvolení programovacího jazyka a vývojového prostředí, v němž bude služba podle návrhu fyzicky vytvořena. Jako velice vhodná volba vývojové platformy se mi jeví J2EE, protože v jazyku java jsou naprogramovány obě aplikace CDEP i SPC a v současné době se jedná společně s platformou .NET o jednu ze dvou primárních platform, které se používají k vývoji servisně orientovaných podnikových řešení (jak uvádí Erl, 2009).

Následuje fáze testování, kdy bude služba v ERDC nasazena do testovacího provozu, kde by měly být odhaleny chyby, které budou následně odstraněny. Rovněž je třeba jasně říci, jestli navržená služba splňuje všechny vstupní požadavky.

Až služba úspěšně projde testovacím procesem, bude nasazena do „ostrého“ provozu, kdy bude plnit svůj účel – rozhraní pro přístup k datům.

Navržená služba ve své podstatě řeší problém společnosti ON Semiconductor se dvojím ukládáním výrobních dat, jelikož odstraňuje potřebu dalšího provozu databáze Synergy. Služba Komunikace vystupuje v roli rozhraní pro přístup k datům v datovém skladu dataPOWER™ pro výběr specifických data, která vyžadují uživatelé aplikací CDEP a SPC. Tím, že se odstraní redundance dat dochází i k efektivnějšímu využívání zdrojů společnosti a úspoře práce i nákladů. Pro uživatele aplikací je přínos v tom, že budou mít k dispozici vždy správná a aktuální data.

Závěr

Cíle této diplomové práce byly dva – rozebrání a jasné vysvětlení pojmu servisně orientovaná architektura a aplikování návrhu implementace SOA v informačním systému společnosti ON Semiconductor.

První kapitola je zaměřena na charakteristiku společnosti ON Semiconductor, je nastíněna historie a odvětví podnikání společnosti.

Druhá kapitola s názvem Teoretická východiska návrhu servisně orientované architektury se detailně zabývá rozborem pojmu servisně orientovaná architektura. Nastínil jsem historie jejího vývoje, její základní vlastnosti a komponenty. Dále je pak popsán životní cyklus zavádění servisně orientovaných řešení, kde jsem detailněji popsal servisně orientovanou analýzu a návrh.

V další kapitole s názvem Analýza současného stavu jsem popsal a zanalyzoval současný stav subsystému informačního systému společnosti ON Semiconductor, v němž jsem řešil problém. V následující kapitole Návrh architektury podle servisně orientovaného přístupu jsem provedl servisně orientovanou analýzu a návrh, jejichž výsledkem je WSDL specifikace služby Komunikace.

Lze konstatovat, že oba cíle této práce jsem splnil – v teoretické části jsem rozebral pojem servisně orientovaná architektura a v praktické části jsem servisně orientovanou architekturu aplikoval v podobě návrhu služby v subsystému informačního systému společnosti ON Semiconductor. Konkrétním přínosem výsledku této práce je dokončená fáze návrhu služby, který je připraven na přezkoumání zaměstnanci ERDC a na jehož základě je možné v budoucnu postavit servisně orientované řešení přístupu rozhraní přístupu k datům v databázi dataPOWER™.

Seznam použité literatury

ARSANJANI, Ali. 2004. *IBM* [online]. 9.11.2004 [cit. 2011-04-19]. Service-oriented modeling and architecture. Dostupné z WWW: <<http://www.ibm.com/developerworks/library/ws-soa-design1/>>.

BALTUS, Jan. 2010a. A přece se vyrábějí : Příběh firmy, která lehkomylně nezhodila lidský potenciál. *Technický týdeník*. 2010, 4, s. 32.

BALTUS, Jan. 2010b. A přece se vyrábějí : Příběh firmy, která lehkomylně nezhodila lidský potenciál. *Technický týdeník*. 2010, 5, s. 32.

BARRY, Douglas. 2011. *Web Services and Service-Oriented Architectures* [online]. 2011 [cit. 2011-04-19]. Service-oriented architecture (SOA) definition. Dostupné z WWW: <http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html>.

ERL, Thomas. 2009. *SOA Servisně orientovaná architektura : Kompletní průvodce*. Vydání první. Brno : Computer Press, a.s., 2009. 671 s. ISBN 978-80-251-1886-3.

FERRIS, Christopher; KARMARKAR, Anish; YUNDLURI, Prasad. 2007. *WS-I : Web Services Interoperability Organization* [online]. 25.10.2007 [cit. 2011-03-02]. Basic Profile Version 2.0. Dostupné z WWW: <[http://www.ws-i.org/Profiles/BasicProfile-2_0\(WGD\).html](http://www.ws-i.org/Profiles/BasicProfile-2_0(WGD).html)>.

GUDGIN, Martin, et al. 2007. *World Wide Web Consortium (W3C)* [online]. 27.4.2007 [cit. 2011-03-06]. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). Dostupné z WWW: <<http://www.w3.org/TR/soap12-part1/>>.

HURWITZ, Judith, et al. 2009. *Service Oriented Architecture For Dummies*. 2nd IBM Limited Edition. Hoboken : Wiley Publishing, Inc., 2009. 62 s. ISBN 978-0-470-52549-4.

CHINNICI, Roberto, et al. 2007. *World Wide Web Consortium (W3C)* [online]. 26.6.2007 [cit. 2011-03-05]. Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. Dostupné z WWW: <<http://www.w3.org/TR/wsdl20/>>.

KALUŽA, Jindřich. 2010. *Informační systémy pro strategické řízení*. Ostrava : VŠB - TU Ostrava, 2010. 144 s. ISBN 978-80-248-2280-8.

WEISS, Petr; RYCHLÝ, Marek. 2007. *Architektura orientovaná na služby, návrh orientovaný na služby, webové služby*. Brno : Vysoké učení technické v Brně, 2007. 34 s.

SINGH, Shaleish. 2005. *SAP Techies : SAP SUPPORT PORTAL* [online]. 5.10.2005 [cit. 2011-04-13]. Web Services and SOA - History. Dostupné z WWW: <<http://www.saptechies.com/web-services-and-soa-history/>>.

DataPOWER Data Dictionary : Release 7.3.6 and Higher. San Jose, California, USA : [s.n.], 2007. 466 s.

Microsoft BizTalk Server [online]. 1.12.2006 [cit. 2011-04-19]. Learn About Service Oriented Architecture (SOA). Dostupné z WWW: <<http://www.microsoft.com/biztalk/solutions/soa/overview.mspx>>.

PDF Solutions : Solutions - By Product - dataPOWER™ [online]. 2008 [cit. 2011-03-30]. Dostupné z WWW: <<http://www.pdf.com/dataPOWER™>>.

The Open Group. *SOA Source Book* [online]. 3rd edition. [s.l.] : Van Haren Publishing, 2010 [cit. 2011-04-13]. Dostupné z WWW: <<http://www.opengroup.org/soa/source-book/intro/index.htm>>.

Seznam zkratek

CDC – Czech Design Center

CDEP – Core Device Electrical Parameters

DVD – Digital Versatile Disc

ERDC – European Region Development Center

GPS – Global Positioning System

J2EE – Java 2 Enterprise Edition

LCD – Liquid Crystal Display

pgc – program class

QoS – Quality of Service

SGML - Standard Generalized Markup Language

SOA – servisně orientovaná architektura

SOAP – Simple Object Access Protocol

SOE – Service Oriented Enterprise

SPC – Statistic Process Control

URI – Uniform Resource Identifier

w3c – World Wide Web Consortium

WSDL – Web Service Definition Language

WS-I – Web Services Interoperability

XML – Extensible Markup Language

XSD – XML Schema Definition

Prohlášení o využití výsledků diplomové práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, diplomovou práci užít (§ 35 odst. 3);
- souhlasím s tím, že diplomová práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že bibliografické údaje o diplomové práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, diplomovou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne

.....

jméno a příjmení studenta

Adresa trvalého pobytu studenta:

.....

Seznam příloh

Příloha 1 – dataPOWER™ extrakční procedura